What is Fault-tolerant Control

Blanke, M.; Frei, C.; Kraus, F.; Patton, R.J.; Staroswiecki, M

Publication date:
2000

Document Version
Også kaldet Forlagets PDF

What is Fault-tolerant Control?

M. Blanke[1], C. Frei[2],
F. Kraus[2], R. J. Patton[3], M. Staroswiecki[4]

[1]Institute of Automation, Technical University of Denmark,
[2]ETH,Zurich,Schwitzerland,
[3]University of Hull, UK
[4]University of Lille, France

Abstract

Faults in automated processes will often cause undesired reactions and shut-down of a controlled plant, and the consequences could be damage to technical parts of the plant, to personnel or the environment. Fault-tolerant control is the synonym for a set of recent techniques that were developed to increase plant availability and reduce the risk of safety hazards. The aim is to prevent that simple faults develop into serious failure. Fault-tolerant control merges several disciplines into a common framework to achieve these goals. The desired features are obtained through on-line fault diagnosis, automatic condition assessment and calculation of appropriate remedial actions to avoid certain consequences of a fault. The envelope of the possible remedial actions is very wide. Appropriate re-tuning can sometimes suffice. In other cases, accommodation of the fault could be achieved by replacing a measurement from a faulty sensor by an estimate. In some situations, complex reconfiguration with on-line controller redesign is required. This paper introduces tools to analyze and explore structure and other fundamental properties of an automated system such that any inherent redundancy in the controlled process can be fully utilized to maintain availability, even though faults may occur.

1 Introduction

Automated systems are vulnerable to faults. Defects in sensors, actuators, in the process itself, or within the controller, can be amplified by the closed-loop control systems, and faults can develop into malfunction of the loop. The closed-loop may alternatively hide a fault from being observed until a situation is reached in which a failure is inevitable. Alternatively, the closed-loop control action may hide a fault from being observed. A situation is reached in which a fault eventually develops into a state where loop-failure is inevitable. A control-loop failure will easily cause production to stop or system malfunction at a plant level.

With economic demand for high plant availability, and an increasing awareness about the risks associated with system malfunction, dependability is becoming an essential concern in industrial automation. A cost-effective way

to obtain increased dependability in automated systems is to introduce fault-tolerant control (FTC). This is an emerging area in automatic control where several disciplines and system-theoretic issues are combined to obtain a unique functionality. A key issue is that local faults are prevented from developing into failures that can stop production or cause safety hazards.

Automation for safety-critical applications, where no failure could be tolerated, requires redundant hardware to facilitate fault recovery. *Fail-operational* systems are made insensitive to any single point component failure. *Fail-safe* systems make controlled shut-down to a safe state when a sensor measurement indicates a critical fault. In contrast, *fault-tolerant* control systems, employ redundancy in the plant and its automation system to make "intelligent" software that monitors behavior of components and function blocks. Faults are isolated, and appropriate remedial actions taken to prevent that faults develop into critical failures. The overall FTC strategy is to keep plant availability and accept reduced performance when critical faults occur.

One way of achieving fault-tolerance is to employ fault diagnosis schemes on-line. A discrete event signal to a supervisor-agent is generated when a fault is detected. This, in turn activates accommodation actions [4], which can be pre-determined for each type of critical fault or obtained from real-time analysis and optimization.

Systematic analysis of fault propagation [1], [6] was shown to be an essential tool for determination of severity of fault effects and for assessment of remedial actions early in the design phase. A semantics for services based on generic component models was developed in [38], [13] and a graphic analysis was found to be very useful. The properties of combined fault diagnosis and control were treated in [31]. [39] focus on the use of fault estimation within a reliable control framework, using the definitions given by [41]. The methods for re-configuration design are comparatively new within the FTC domain. A few schemes have come into real application. Predetermined design for accommodation was demonstrated for a small satellite in [7], and [6]. Techniques using logic inference on qualitative models were used in [26] and [27].

A related area is the control of discrete-event dynamical systems (DEDS) which have a known structure with pre-determined events that occur with unknown instants and sequence [43]. Diagnosis for DEDS was treated in [33] and [25]. DEDS is a sub-class of FTC, where events are not pre-determined and system structure can change when faults occur.

Concerning implementation, a correct and consistent control system analysis should always be followed by equally correct software implementation. This is particularly relevant for the supervisory parts of an FTC scheme [19]. Testing of the FTC elements is difficult since it is difficult to replicate the real conditions under which faults occur. Well planned software architecture and implementation are thus crucial issues for FTC implementation. A study of the use of object-oriented programming architectures was described by [24]. The FTC area is hence very wide and involves several areas of system theory. One overview [31] emphasized many algorithmic essentials and the role of FDI.

Another [4] presented an engineering view of the means to obtain FTC.

This paper focus on the methodological issues in analysis for FTC. The severity of faults is first addressed through analysis of fault propagation, which provides a list of faults that should be stopped from developing into failure due to the severity of their end effects. The possibilities to detect and stop propagation of particular faults are then dealt with. A structural analysis technique is introduced, which uses graph theory to determine which redundancy exist in the system and thus shows the possibilities to diagnose and handle particular faults [13], [34], [19]. The structure of the problem gives a number of possibilities for recovery. The ability to control or observe the system [22], [42] are extended to measure these properties of a system after a particular fault [12]. Next, implementation of a fault-tolerant control scheme is proposed as a layered structure where an autonomous supervisor implements detection and reconfiguration using the necessary logics. The overall development strategy is finally summarized and an example illustrate features of the methods.

2 Basic Definitions

As this is a new engineering field terminology it is particularly important to define the terminology carefully. A short list is enclosed with the main terms. A longer list can be found in the IFAC - SAFEPROCESS terminology definition [17]. In addition to terminology, different control methods should be clearly distinguished. Definitions are included to specify explicitly what should be understood by the term fault-tolerant control.

2.1 Terminology

- *Constraint :* a functional relation between variables and parameters of a system. Constraints may be specified in different forms, including linear and nonlinear differential equations, and tabular relations with logic conditions between variables.

- *Fail-operational* : a system is able to operate with no change in objectives or performance despite of any single failure.

- *Fail-safe* : a system fails to a state that is considered safe in the particular context.

- *Fault-tolerance* : the ability of a controlled system to maintain control objectives, despite the occurrence of a fault. A degradation of control performance may be accepted. Fault-tolerance can be obtained through fault accommodation or through system and /or controller reconfiguration.

- *Fault-accommodation* : change in controller parameters or structure to avoid the consequences of a fault. The input-output between controller

and plant is unchanged. The original control objective is achieved although performance may degrade.

- *Reconfiguration :* change in input-output between the controller and plant through change of controller structure and parameters. The original control objective is achieved although performance may degrade.

- *Supervision :* the ability to monitor whether control objectives are met. If not, obtain/calculate a revised control objective and a new control structure and parameters that make a faulty closed loop system meet the new modified objective. Supervision should take effect if faults occur and it is not possible to meet the original control objective within the fault-tolerant scheme.

- S*tructure graph :* A directed graph representing the general dynamic equations (constraints) that describe the system. Constraints that are isomorphic mappings are denoted by double arrows on arcs. Non-isomorphic mappings are indicated by unidirectional arcs in the graph. The graph has the special property that it is bipartite.

2.2 Definitions

A standard control problem is defined by a control objective O, a class of control laws $\mathbb{U}$, and a set of constraints C. Constraints are functional relations that describe the behavior of a dynamic system. Linear or nonlinear differential equations constitute very useful representations of constraints for many physical systems. Other types of models are necessary in other cases. The constraints define a structure S and parameters θ of the system. Solving the control problem means to find in $\mathbb{U}$ a control law U that satisfies C while achieving O . Some performance indicator J could be associated with a control objective O. When several solutions exist, the best one is selected according to J. The control problem is defined as:

Definition 1 *The control problem : Solve the problem* $< O, S, \theta, \mathbb{U} >$ *where the structure* S *and parameters* θ *of the constraints* C *are distinguished.*

Now suppose that we only know the set to which the actual value belongs, e.g. due to time-varying parameters or uncertainty, the control problem is now to achieve O under constraints whose structure is S and whose parameters belong to a set Θ. Two solution approaches can be defined: robust control minimizes the discrepancy over Θ of the achieved results, while adaptive control first estimates the "true" parameter $\hat{\theta}$.

Definition 2 *The robust control problem : Solve* $< O, S, \Theta, \mathbb{U} >$ *where* Θ *stands for a set of possible* θ *values.*

Definition 3 *The adaptive control problem : Solve* $< O, S, \hat{\theta}, \mathbb{U} >$ *where* $\hat{\theta} \in \Theta$ *is estimated as part of the adaptation.*

The next problem extension is $< O, \mathbb{S}, \Theta, \mathbb{U} >$ where $\mathbb{S}$ stand for a given set of constraint structures. Define some deterministic automaton Γ, which shifts from one pair $(S, \theta) \in \mathbb{S} \times \Theta$ to another one (hybrid control). The problem is to achieve O under a sequence of constraints which is defined by Γ. When O itself is decomposed into a sequence of goals, the problem becomes $< \mathbb{O}, \mathbb{S}, \Theta, \mathbb{U} >$ with Γ shifting from one quadruple $(O, S, \theta, U) \in \mathbb{O} \times \mathbb{S} \times \Theta \times \mathbb{U}$ to another.

Next, consider $< O, \mathbb{S}, \Theta, \mathbb{U} >$ with uncertain knowledge about (S, θ). O has to be achieved under constraints whose structure and parameters are partly or fully unknown, except that (S, θ) belongs to $\mathbb{S} \times \Theta$. Let a control objective O and a nominal system (S^*, θ^*) be given. Let (S, θ) be the actual constraints and $(\hat{S}, \hat{\theta})$ the estimated ones. Nominal control obviously solves $< O, S^*, \theta^*, \mathbb{U} >$. When a fault occurs, $(S, \theta) \neq (S^*, \theta^*)$ and nominal control is no longer suitable. This is a generalization of the robust and adaptive control problems. Both the parameters and the structure of constraints may change when faults occur.

Definition 4 *The fault-tolerant control problem: Solve* $< O, \hat{\mathbb{S}}, \hat{\Theta}, \mathbb{U} >$ *where* $\left(\hat{\mathbb{S}}, \hat{\Theta}\right)$ *is the set of possible structures and parameters of the faulty system. Where diagnosis is available, the set* $\left(\hat{\mathbb{S}}, \hat{\Theta}\right)$ *could be provided by a diagnosis task.*

Many different approaches can be used to solve the FTC problem $< O, \hat{\mathbb{S}}, \hat{\Theta}, \mathbb{U} >$ [31]. However, robust approaches, which achieve the goal for any pair (S, θ) are clearly unrealistic in the general case.

We define two subsets of fault-tolerant control, one is accommodation, the other reconfiguration.

Definition 5 *Fault accommodation : Solve the control problem* $< O, \hat{S}, \hat{\theta}, \mathbb{U} >$ *where* $(\hat{S}, \hat{\theta})$ *is the estimate of the actual constraints, e.g. provided by fault diagnosis algorithms.*

The fault(s) can thus be accommodated if $< O, \hat{S}, \hat{\theta}, \mathbb{U} >$ has a solution. If accommodation is not possible, another problem has to be stated, by finding a pair (Σ, τ) among all feasible pairs $\mathbb{S} \times \Theta$, such that $< O, \Sigma, \tau, \mathbb{U} >$ has a solution. A pair (Σ, τ) is considered feasible if it belongs to the fault-free parts of the faulty system. Fault diagnosis may identify a set $\left(\hat{\mathbb{S}}, \hat{\Theta}\right)$ to give an estimate of the constraints of the faulty system. This is not a necessary prerequisite but the availability of such estimate will improve the possibility of finding said solution. This procedure is an active approach, the control is changed as a consequence of our knowledge of the new control problem.

Reconfiguration is hence defined as follows:

Definition 6 *Reconfiguration : Find a new set of system constraints* $(\Sigma, \tau) \in (\mathbb{S}, \Theta) \Big| (\hat{\mathbb{S}}, \hat{\Theta})$ *such that the control problem* $< O, \Sigma, \tau, \mathbb{U} >$ *has a solution. Activate this solution. The choice of a new set of constraints will imply that input-output relations between controller and plant are changed.*

The difference between accommodation and reconfiguration whether input-output (I/O) between controller and plant is changed. Reconfiguration implies use of different I/O relations between the controller and the system. Switch of the system to a different internal structure, to change its mode of operation, is an example of such I/O switching. Accommodation does not use such means.

Both fault accommodation and system reconfiguration strategies may need new control laws in response to faults. They also have to manage transient behavior, which result from the change of control law or change of the constraints' structure.

It is noted that the set of feasible pairs $\mathbb{S} \times \Theta$ may depend on the fault(s). If such a pair does not exist, this means that O can be achieved neither by fault accommodation nor by system reconfiguration. The only possibility is thus to change O.

The most general problem is defined by the triple $< \mathbb{O}, \mathbb{S}, \Theta, \mathbb{U} >$ where $\mathbb{O}$ is a set of possible control objectives. In view of its practical interpretation, $< \mathbb{O}, \mathbb{S}, \Theta, \mathbb{U} >$ is defined as a supervision problem in which the system goal is not pre-defined, but has to be determined at each time taking into account the actual system possibilities.

A supervision problem is thus an FTC problem associated with a decision problem: when faults are such that fault-tolerance cannot be achieved, the system goal itself has to be changed [36]. When far-reaching decisions with respect to the system goal have to be taken, human operators are generally involved.

Definition 7 *Supervision : Monitor the triple* (O, S, θ) *to determine whether the control objective is achieved. If this is not the case, and the fault tolerant problem does not have a solution, then find a relaxed objective* $\Gamma \in \mathbb{O}$ *and a pair* $(\Sigma, \tau) \in \mathbb{S} \times \Theta$, *such that the relaxed control problem* $< \Gamma, \Sigma, \tau, \mathbb{U} >$ *has a solution.*

If no such triple exists, failure is unavoidable. This can be a design error or a deliberate choice to accept certain failure scenarios, e.g. for reasons of cost/benefit or small likelihood for a certain event. The choice of a new objective can be made autonomously in rare cases only. Most commonly, human intervention is needed, using decision support from the diagnosis and overall goals for the plant [23], [37]. It should be noted that the choice of Γ could be made to include the fail-to-safe condition where control is no longer active but plant safety is not at stake.

Definition 8 *A system is recoverable from a fault iff a solution exists to at least one of the problems* $< O, \hat{S}, \hat{\theta}, \mathbb{U} >$ *and* $< O, \Sigma, \tau, \mathbb{U} >$.

Definition 9 *A system is weakly recoverable if a solution exists to* $< \Gamma, \Sigma, \tau, \mathbb{U} >$.

3 Analysis of Fault Propagation

The first step in a fault-tolerant design is to determine which failure modes could severely affect the safety or availability of a plant. Analysis of failure of parts of a system is a classical discipline and the failure mode and effects analysis (FMEA) is widely used and appreciated in industry. The traditional FMEA does not support analysis of the handling of faults, only of their propagation. In automated systems, when the goal of fault-tolerance is to continue operation, if this is at all possible. An extended method for fault propagation analysis (FPA) was hence suggested in [1] using an algebraic approach for propagation analysis. The aim of the FPA is to show end effects of faults, and assist in designing for fault tolerance such that end effects with severe consequences are stopped if the system structure makes this possible. If the FPA analysis finds that serious failure can occur due to certain faults, these are included in a list of fault effects to be detected. Whether this is possible is disclosed in a later analysis of structure that shows which redundant information is available in the system [38], [8], [9]. The final step will then be to find actions, preferably within the software of the controller, that can accommodate the fault and prevent the serious end effects from occurring. Analysis of whether this will be possible is a part of the analysis for the quality measures of recovery for the system [12], [34].

3.1 Fault propagation

For the reasons given above, fault analysis needs to incorporate analysis throughout a system. In order to do this a component-based method was introduced [1], in which possible component faults are identified at an early stage of design. The method uses the FMEA description [21], [16] of components as a starting point. In this context components are sensors, valves, motors, programmable functions etc.. Programmable parts are considered as consisting of separate function blocks that can be treated similarly to physical components in the analysis, bearing in mind that their properties may be changed by software modifications if so desired.

An FMEA scheme shows how fault effects out of the component relate to faults at inputs, outputs, or parts within the components.

Definition 10 *Fault propagation matrix: boolean mapping of component faults* $f_c \in \mathcal{F}$ *onto effects* $e_c \in \mathcal{E}$*:*

$$\mathbf{M}: \quad \mathcal{F} \times \mathcal{E} \curvearrowright \{0,1\}\,; \quad m_{ij} = \begin{cases} 1 & if \quad f_{cj} = 1 \Longrightarrow e_{ci} = 1 \\ 0 & otherwise \end{cases}$$

An FMEA scheme can be expressed as

$$\mathbf{e}_{ci} \leftarrow \mathbf{M}_i^f \otimes \mathbf{f}_{ci} \tag{1}$$

where $\mathbf{M}_i^f$ is a Boolean matrix representing the propagation. The operator $\otimes$ is the inner product disjunction operator that performs the boolean

operation

$$e_{cik} \leftarrow (m_{ik1} \wedge f_{ci1}) \vee (m_{ik2} \wedge f_{ci2}) \ldots \vee (m_{ikn} \wedge f_{cin}) \tag{2}$$

When effects propagate from other components, we get, at level i:

$$\mathbf{e}_{ci} \leftarrow \mathbf{A}_i^f \otimes \begin{bmatrix} \mathbf{f}_{ci} \\ \mathbf{e}_{c(i-1)} \end{bmatrix} \tag{3}$$

This is a surjective mapping from faults to effects: there is a unique path from fault to end effect, but several different faults may cause the same end effect.

System descriptions are obtained from interconnection of component descriptions. Merging two levels gives the end effects at the second level,

$$\mathbf{e}_{c2} \leftarrow \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \left(\mathbf{A}_2^f \otimes \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ 0 & \mathbf{A}_1^f \end{bmatrix} \right) \end{bmatrix} \otimes \begin{bmatrix} \mathbf{f}_{c2} \\ \mathbf{f}_{c1} \end{bmatrix} \tag{4}$$

Eventually, end effects at the system level are reached.

A mapping of observed effects to possible faults are obtained through $\mathbf{f}_c = \left(\mathbf{M}^f\right)^{-1} \mathbf{e}_c$. And $\left(\mathbf{M}^f\right)^{-1} = \mathbf{M}^T$ since $\mathbf{M}^f$ is Boolean. Analysis of the system matrix can easily show where in the system the propagation should be detected and stopped.

When there is no logical feedback involved, the result is the capability of isolation of fault effects at any level. If feedback is involved, we have a principal difficulty: if a cut in the graph of a boolean loop is stable - the two sides of the cut remain equal after one propagation around the loop - then the loop is a tautology and can be eliminated. If not, no boolean solution exists. A "dedicated loop treatment" was employed in [3] to define the two sides of a loop cut as additional input and output, and leave the further judgement to the designer. This shortcoming is shared by several methods in reliability engineering. The principal difficulty is caused by the binary modeling of faults and their propagation.

With this obstacle in mind, fault propagation analysis should be the first step in a fault-tolerant design. The systematic approach forced upon the designer should not be underestimated, and might even be an asset as an essential part of the safety assessment needed in many industrial designs.

Experience from applying fault propagation analysis to larger systems show that we might need to include occurrence of one fault and the non-occurrence of another in the description [6]. This means extending $\mathbf{f}_i$ to $\left[\mathbf{f}_i, \bar{\mathbf{f}}_i\right]^T$ in the above expressions.

4 Structural Analysis

The structural model of a system, see [35] and [10], is a directed graph that represents the relations between system variables and parameters (known and

unknown), and the dynamic equations (constraints) that describe the system behavior. Analysis of the system-structure graph will reveal any system redundancy, and particular sub-systems can be identified which can be exploited to obtain fault-tolerance.

4.1 Structural model

Let $F = \{f_1, f_2, ..f_m\}$ be the set of the constraints which represent the system model and $Z = \{z_1, z_2, ..z_n\}$ the set of the variables and parameters. With K the subset of the known and X the subset of the unknown elements in Z, $Z = K \cup X$. Z is allowed to contain time derivatives, so that dynamic systems as well as static ones can be described by their structure.

Definition 11 *The structure graph of a system is a bipartite graph* (F, Z, A) *where elements in the set of arcs* $A \subset F \times Z$ *are defined by :* $(f_i, z_j) \in A$ *iff the constraint* f_i *applies to the variable or parameter* z_j, *(* $f_i \in F$ *with* $i = 1, ...m$ *and* $z_j \in Z$ *with* $j = 1, ...n$*).*

The structure-graph is thus a directed graph and it has the property of being bipartite . A graph is bipartite (see [15]) if its vertices can be separated into two disjoint sets F and Z in such a way that every edge has one endpoint in F and the other in Z.

Definition 12 *A sub-system is a pair* $(\phi, Q(\phi))$, *where* $\phi \in \mathcal{P}(F)$ *is a subset of* F, *and* Q *is defined by*

$$\begin{aligned} Q: \quad & \mathcal{P}(F) \rightarrow \mathcal{P}(Z); \\ & \phi \rightarrow Q(\phi) = \{z_j | \quad \exists f_i \in \phi \text{ such that } (f_i, z_j) \in A\} \end{aligned}$$

In this definition, a sub-system is any subset of the system constraints ϕ along with the related variables $Q(\phi) \in Z$. There are no specific requirements to the choice of the elements in ϕ. $\mathcal{P}(F)$ is the set of the subsets of F and it contains all possible sub-systems. The sub-graph that is related to a sub-system is the structure of the sub-system, $(\phi, Q(\phi))$.

Sub-systems are used to find alternate paths through a structure-graph. When a fault occur we need to exploit alternative ways to access affected variable(s) $x_f \in X$ than through the faulty connection f_f .The design task is to express x_f through alternative subsystem(s) and eventually through associated known variable(s), in K. This means to exploit analytic redundancy relations in the system.

4.2 Matching on a structure-graph and canonical decomposition

The set of constraints is separated in F_K, those that apply only to known variables, and F_X , which apply to unknown elements in Z, $F = F_K \cup F_X$. We are interested in the analysis of the sub-graph $G(F_X, X, A_X)$ in order

to determine which analytic redundancy relations exist that can help access a particular unknown variable. If redundant sub-graphs are available, then the particular variable could be observed or controlled via the redundant path should a fault occur in the other.

Definition 13 *Let* $a \in A_X, F(a) \in F_X$ *and* $X(a) \in X$, *such that* $a = (F(a), X(a))$. *A matching* M *is a subset of* A_X *such that :*
$\forall a$ *and* $b \in M, F(a) \neq F(b)$ *and* $X(a) \neq X(b)$.
A complete matching on F_X*:* $\forall f \in F_X \quad \exists x \in X$ *such that* $(f, x) \in M$
A complete matching on X*:* $\forall x \in X \quad \exists f \in F_X$ *such that* $(f, x) \in M$.

Theorem 1 [11] *Any bipartite graph of finite external dimension can be uniquely decomposed into three sub-graphs:*

- $G^+ = (F^+, X^+, A^+)$ *such that* $Q(F^+) = X^+$*and a complete matching exists on* X^+ *but not on* F^+.
- $G^= = (F^=, X^=, A^=)$ *such that* $Q(F^=) = X^= \cup X^+$ *and a complete matching exists on* $X^=$ *as well as on* $F^=$.
- $G^- = (F^-, X^-, A^-)$ *such that* $Q(F^-) = X^- \cup X^= \cup X^+$ *and a complete matching exists on* F^- *but not on* X^-.

4.3 Interpretation for diagnosis

- G^+ is the over-constrained (redundant) sub-system, i.e. $|F^+| > |X^+|$. For solutions X^+ to exist, compatibility conditions have to hold, which are the system analytic redundancy relations (ARR). They can be expressed using only the known system variables. The variables X^+ are thus accessible and the system components whose constraints belong to $F_K \cup F^+$ are monitorable.
- $G^=$ is the just-constrained sub-system, i.e. $|F^=| = |X^=|$. $X^=$ can be computed as a function of the known variables, but no redundancy exists. The variables $X^=$ are observable but the system components whose constraints belong to $F^=$ are not monitorable.
- G^- is the under-constrained sub-system, and $|F^-| < |X^-|$. There is not enough equations for the computation of X^-. Neither are the variables X^- observable nor are the components whose constraints belong to F^- monitorable.

4.4 Redundancy Relations and Causality

From the matching definition, each pair $a = (F(a), X(a))$ can be interpreted as follows : $X(a)$ is a consequence of the constraint $F(a)$ in which all the variables $Q(F(a))$ except $X(a)$ would have values imposed from the rest of the system and the environment. However, not all matching can receive this

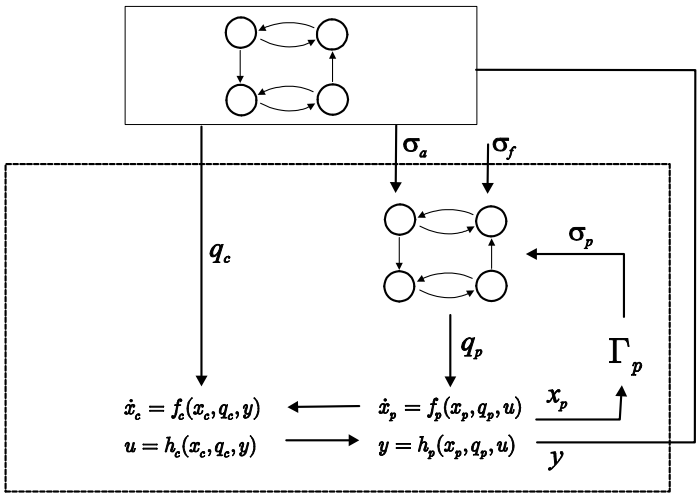

Figure 1: Fault tolerant control systems are hybrid by nature. The dynamics of the plant is influenced by mode changes q_p, the controller by q_c. σ_f denote fault events, σ_a control events. Plant internal events are σ_p.

nice interpretation, and the set of the possible causality assignments (causal matching) is only a sub-set of the possible matching. Using causal matching, the analytic redundancy relations appear on the structural graph as alternated chains, making it easy to design the parity relations used for fault diagnosis. Moreover, a causal graph can be associated with each causal matching on a structural graph , providing structural tools for alarm filtering and reconfigurability analysis [34].

The structural approach can thus treat the all issues of principal interest regarding redundancy for sensing and actuation possibilities in a plant.

5 Recoverability

A fault is a discrete event that acts on a system and by that changes some of the properties of the system. The goal of fault-tolerant control is in turn to respond to the occurrence of a fault such that the faulty system still is well behaved. This is achieved by accommodation of the fault or by reconfiguration. Due to these discrete nature of fault occurrence and reconfiguration, FTC systems are hybrid in nature. This is illustrated in figure 1 where σ_f denotes fault events, σ_a denotes control events reconfiguring the system and q_c denotes the control mode which selects a control law. The actual physical mode q_p of the plant my be viewed as the discrete state of an automaton which is driven by plant internal events σ_p, the fault events σ_f and the control events σ_a.

The analysis of the behavior of fault tolerant control system is not trivial

[12]. For the design of fault tolerant control systems the hybrid nature is usually neglected and the focus lies with fault detector design and selection of remedial action . A popular example in case of a sensor fault is to reconstruct the value of a faulty sensor by means of an observer an leave the actual control law unchanged (see e.g. [40], [5]).

5.1 Quality Measures for Recovery

However, if we neglect the time and behavior of the system between fault occurrence and system and/or controller reconfiguration, it is easy to see that whether a system can be recovered from a fault or not is not primarily a question of clever controller design but of the properties of the reconfigured faulty system. Moreover, the extent to which the functionality of a system can be recovered from a fault depends on how much control over the system is still available and how much state or output information can be still obtained.

5.1.1 Quality indicators

The reconfiguration problem was defined as the solution of the control problem $< O, \Sigma, \tau, \mathbb{U} >$ where $(\Sigma, \tau) \in \left((\mathbb{S}, \Theta) \middle| (\hat{\mathbb{S}}, \hat{\Theta})\right)$. The control problem is solved by satisfying the control objective, often expressed through an indicator J. Different structural alternatives can then be compared through the achievable value of the indicator associated with the particular structure,

$$J_x(\Sigma, \tau) = \min_{u \in \mathbb{U}} J(O, \Sigma, \tau, \mathbb{U}) \tag{5}$$

If a limit J_a exists for solutions to be admissible, $\mathbb{J} = \{J_x(\Sigma, \tau) | J_x(\Sigma, \tau) < J_a\}$, then the cardinality of $\mathbb{J}$ represents the number of admissible reconfiguration solutions, selected out of a possibly larger set that has the necessary structural properties. In model predictive control, the optimal J is found and the associated controller selected for use after fault diagnosis has provided an assessment of the structure and parameters of the faulty system.

The calculation of J_x may be quite heavy since the complete closed loop optimization problem is solved for each of the possible reconfigurations. Measures of the control energy needed to change a particular state from one value to another could also be useful, and simpler to calculate. A similar measure of output observation could express the ease with which measurements could be reconstructed. This argumentation leads to the definition of a quality measure for recovery using the underlying system properties controllability and observability.

Consider a set of systems $S(q_f, q_c)$ parameterized by the configurators q_f and q_c (i.e. $q_p = (q_a, q_f)$. That is $S(\varnothing, \varnothing)$ represents the nominal fault free system, $S(q_f, \varnothing)$ represents the system after occurrence of the fault event and $S(q_f, q_c)$ denotes the faulty system after accommodation/reconfiguration.

5.1.2 Measures based on Gramians

Linear measures of quality require a linearized system description. Consider therefore the linear time invariant (faulty) system $S(q_c, q_f)$ given by

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}(q_c, q_f)\mathbf{x}(t) + \mathbf{B}(q_c, q_f)\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}(q_c, q_f)\mathbf{x}(t) + \mathbf{D}(q_c, q_f)\mathbf{u}(t) \end{aligned} \tag{6}$$

Let us more closely study sensor and actuator faults. Determining the influence of a missing faulty sensor or actuator on the systems's operability is a question also studied in connection with the selection of actuators and sensors see e.g. [20]). While for the classical sensor/actuator selection problem the goal is to identify and remove those of minor or nearly identical influence. Fault tolerant control aims at retaining some of the redundant sensors and actuators. [30] utilize measures for the degree of observability. and controllability. These measures may also be utilized to be an indicator for the quality of the recoverability. The measures of choice are the observability gramian

$$\mathbf{W}_o(S) = \int_0^\infty e^{\mathbf{A}^T t}\mathbf{C}^T\mathbf{C}e^{\mathbf{A}t}dt \tag{7}$$

and similarly for the controllability

$$\mathbf{W}_c(S) = \int_0^\infty \mathbf{B}^T e^{\mathbf{A}^T} e^{\mathbf{A}t}\mathbf{B}dt \tag{8}$$

Gramians which allow to identify direction in state space of different degree of controllability and observability (assuming an adequate non-dimensionalisation of the system's states, inputs, and outputs). This means, for some non-dimensionalised state x_0 , the quantity $x_0^T W_o x_0$ represents the observation "energy" obtained from this state. For any x, which is a unit length eigenvector, the obtained observation energy is determined by the corresponding eigenvalue. An unobservable direction provides zero observation energy. The fact that the matrix determinant combines information about all eigenvalues, motivates the following quality indicator for measurement recovery

$$\rho_o(q_c, q_f) = \sqrt[n]{\frac{|\mathbf{W}_o(S(q_c, q_f))|}{|\mathbf{W}_o(S(\varnothing, \varnothing))|}} \tag{9}$$

where the n^{th} root serves to make the measure independent of the system dimension n.

For illustration, consider a system with two equivalent sensors. If one sensor fails, and operation is continued with a single sensor, $\rho_o(q_c, q_f) = \frac{1}{2}$. In this deterministic framework, there is no difference between the two sensors. The value $\frac{1}{2}$indicates a harder state estimation problem. Note that $\rho_o(q_c, q_f) = 0$ if the system is not recoverable from the fault q_f by the reconfiguration q_c.

Similar arguments lead to the definition

$$\rho_c(q_c, q_f) = \sqrt[n]{\frac{|\mathbf{W}_c(S(q_c, q_f))|}{|\mathbf{W}_c(S(\varnothing, \varnothing))|}} \tag{10}$$

as a quality indicator for control recovery after an actuator fault.

Both measures Eq.10 and Eq.9 assume that $S(\varnothing, \varnothing)$ is controllable and observable. If this is not the case, the measure should be applied to the observable or controllable subspaces, only.

The consequence of a zero measure of Eq.10 is that a closed-loop observer can not be designed. Nevertheless, an open-loop solution may still exist for use as a short-term replacement signal for a failed sensor , [2].

5.2 Combined Analysis

The above measures allow to assess a system's recoverability for a specific situation. Especially during the design phase of a FTC system a structural analysis (see also [34]) may help to find suitable locations for redundant sensors or actuators. Here too, we encounter questions very closely related to question in the design of control systems (see e.g. [29]).

The structure of a system (Eq.6) may be represented by a graph or a structural matrix [22] as shown in Figure 2. A system $(\mathbf{A}(q_c, q_f), \mathbf{B}(q_c, q_f))$ is structurally controllable iff [14]

1. each state node is accessible from at least one control node
2. the generic rank of the structural matrix $(\mathbf{A}(q_c, q_f)\ \mathbf{B}(q_c, q_f))$ is n.

Determining structural observability is the dual problem. A system is thus (structurally) recoverable if it remains structurally controllable and structurally observable. Consider the example in Figure 2. This system is (structurally) recoverable from fault in actuator u_2 but not in u_1 This is determined from the structural controlability. With a fault in u_2,we get the rank = 2 matrix

$$(A, B) = \left(\begin{array}{cccc} x & x & 0 & 0 \\ x & x & 0 & , \ 0 \\ x & x & x & x \end{array} \right) \tag{11}$$

whereas the fault in $u_1$gives a matrix with rank = 3, thus shows structural controllability of the system,

$$(A, B) = \left(\begin{array}{cccc} x & x & 0 & x \\ x & x & 0 & , \ 0 \\ x & x & x & 0 \end{array} \right) \tag{12}$$

Note that structural recoverability provides a stronger negative result then above measures in the sense that if a system is (structurally) non recoverable from a fault then it is non recoverable for any realization of the system $(\mathbf{A}(q_c, q_f), \mathbf{B}(q_c, q_f))$. By this it is well suited for ruling out ineffective redundancy alternatives.

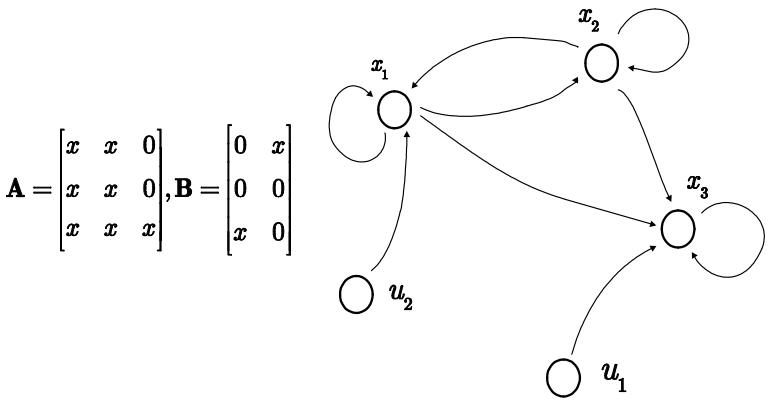

Figure 2: Structural representations of a dynamical system are traditionally used for studying structural observability and controllability. They can also be used to assess structural recoverability when we know how a fault affects A and B.

5.2.1 Functional controllability

For many control applications, state controllability and observability are not of prime importance and one is rather concerned with controllability characterization of the input/output. Functional controllability [32] addresses this question. A system is called functional controllable if for a "suitable" output vector $y(t)$ defined for $t > 0$ there exists a vector $u(t)$ which generates the output vector $y(t)$ from the initial condition $x = 0$ where suitable refers to some smoothness and Laplace transformability conditions.

A system with the transfer function $G(s)$ is functionally controllable iff

1. $|\tilde{G}(s)| \neq 0$

where $\tilde{G}(s)$ is a part of $G(s)$ of dimension $\dim(\mathbf{y}(t)) \times \dim(\mathbf{y}(t))$. Note that this implies $\dim(\mathbf{u}(t)) \geq \dim(\mathbf{y}(t))$. Clearly the system is functionally recoverable from the fault q_f by the reconfiguration q_c iff there exists a $G_{q_c,q_f}(s)$ such that $|\tilde{G}_{q_c,q_f}(s)| \neq 0$ of dimension $\dim(\mathbf{y}(t)) \times \dim(\mathbf{y}(t))$.

5.3 Consequences for FTC

In this section we will show that this recoverability concept is indeed a system property with immediate consequences for fault tolerant control.

5.3.1 Reconstruction of measurements

A strategy which is sometimes considered as a remedial action to a sensor fault is to reconstruct the missing measurement, using this instead of the

original measurement with the existing control law [40] and [5]. However, this might not always be possible. First, it must be possible to reconstruct the missing measurement from the remaining measurements. Otherwise, the strategy leads to open loop control of certain modes, which could only be a short time remedial action.

A measurement of a faulty sensor can be reconstructed from the remaining measurements if and only if the system is recoverable from that sensor fault. To show this, consider the generalized eigenvector decomposition of the system and write:

$$y_i(t) = \mathbf{c}_i x(t) = \mathbf{c}_i \sum_{j=1}^{n} \xi_j(t)\mathbf{q}_j = \sum_{j=1}^{n} \xi_j(t)\mathbf{c}_i\mathbf{q}_j$$

where y_i is the measurement to be reconstructed, $\mathbf{q}_j$ are the eigenvector directions of the system, $\xi_j(t)$ the time evolution along these directions and n the dimension of the state space of the system.

To show that recoverability is sufficient note that all directions $\mathbf{q}_j$ for which $\mathbf{c}_i\mathbf{q}_j \neq 0$ contribute to the measurement y_i. If the system is recoverable from the failure of sensor i these directions remain observable and thus $y_i(t)$ can be reconstructed.

On the other hand: If the system is not recoverable it looses observability (sensor fault). The direction(s) $\mathbf{q}_j$ for which observability is lost was observable in combination with sensor i and thus $\mathbf{c}_i\mathbf{q}_j \neq 0$. That is the unobservable direction $\mathbf{q}_j$ would be needed to reconstruct the output y_i.

This basically means that there are directions that are only observable in combination with the output y_i and therefore can not be reconstructed if y_i is missing.

6 Autonomous Supervision

Autonomous supervision requires development and implementation observing completeness and correctness qualities. It is important that the design of a supervised control system follows a modular approach, where each functionality can be designed, implemented, and tested independently of the remaining system. The algorithms that realize the supervisory functionality constitute themselves an increased risk for failures in software, so the overall reliability can only be improved if the supervisory level is absolutely trustworthy. General design principles were treated in [4], development methods were improved and an implementation demonstrated in a satellite application in [6]. A seven-step design procedure was shown to lead to an significantly improved logic design compared to what was obtainable by conventional ad-hoc methods. The design of the autonomous supervisor was the subject in [19] where the COSY ship propulsion benchmark was the main example [18]. A software architecture for fault-tolerant process control was suggested in [24].

The experience from the above studies was that design of an autonomous supervisor relies heavily on having an appropriate architecture that supports clear allocation of methods to different software tasks. This is crucial for both development and verification. The latter is vital since test of the supervisor functions in an autonomous control system is a daunting task.

6.1 Architecture

The implementation of a supervisory level onto a control system is not a trivial task. The architecture shall accommodate the implementation of diverse functions

- Support of overall coordinated plant control in different phases of the controlled process; start-up, normal operation, batch processing, event triggered operation with different control objectives, close-down.
- Support of all control modes for normal operation and modes of operation with foreseeable faults.
- Autonomous monitoring of operational status, control errors, process status and conditions.
- Fault diagnosis, accommodation and re-configuration as needed. This is done autonomously, with status information to plant-wide coordinated control.

These functions are adequately implemented in a supervisory structure with three levels in the autonomous controller, and communication to a plant-wide control as the fourth. The autonomous supervision is composed of levels 2 and 3, taking care of fault diagnosis, logic for state control and effectors for activation or calculation of appropriate remedial actions. This is illustrated in Figure 3 that shows:

1. A lower level with input/output and the control loop.
2. A second level with algorithms for fault diagnosis and effectors to fault accommodation.
3. A third level with supervisor logic.
4. A fourth layer with plant-wide control and co-ordination.

The Control Level is designed and tested in each individual mode that is specified by different operational phases and different instrumentation configurations. The miscellaneous controller modes are considered separately and it is left to the supervisor design to guarantee selection of the correct mode in different situations.

The Detectors are signal processing units that observe the system and compares with the expected system behavior. An alarm is raised when an anomaly is detected. The Effectors execute the remedial actions associated with fault accommodation/reconfiguration.

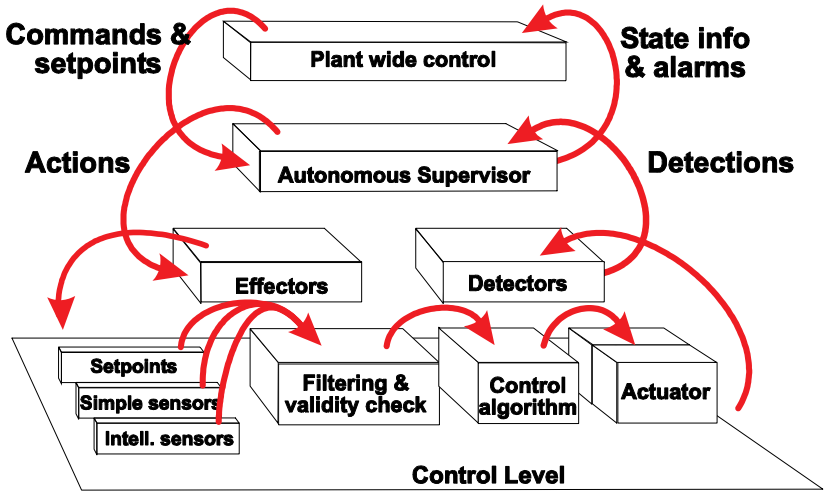

Figure 3: Autonomous supervisor comprises fault diagnosis, supervisor locic and effectors, the latter to cary out the necessary remedial actions when faults are diagnosed. The upper level is plant-wide control and operator supervision.

6.2 Design Procedure

When the level of autonomy becomes high and thereby demands a higher level of reliable operation, it becomes inherently more complex for the designer to cover all possible situations and guarantee correct and complete operation [28], [6].

A systematic design strategy will use the analysis of fault propagation and structure as basic elements:

1. *Fault propagation:* A Fault Propagation Analysis of all relevant subsystems is performed and combined into a complete analysis of the controlled system. The end-effects describe consequences on top level. The FMEA schemes for components is used as a basis to facilitate reuse of accumulated knowledge about faults and failures.

2. *Severity assessment*: The top level end-effects are judged for severity. The ones with significant influence on control performance, safety or availability are selected for treatment by the autonomous supervisor. A reverse deduction of the fault propagation is performed to locate the faults that cause severe end-effects. This gives a short-list of faults that should be detected.

3. *Structural analysis*: System structure is analyzed for each of the short-listed faults from step 2. The graph method gives a "yes-no" type of

information whether sufficient redundancy is available in the system to detect each of the selected faults.

4. *Possibilities for FTC*: The possibilities to obtain fault-tolerance are considered. For each of the short-listed faults, this means utilize physical redundancy, then analytical redundancy. Use the measures of recovery quality in listing the most promising candidates for accommodation/recovery. Whether the original control objectives can be met will not be known at this stage of design.

5. *Select remedial actions*: The possibilities in 4 are further elaborated. Look into enabling and disabling redundant units, select among possible accommodation or reconfiguration actions. If the original control objectives can not be met, handling of the problem by a the supervision function must be considered. The autonomous part of supervision must always be to offer graceful degradation and close down when this is necessary as fall-back.
   The remedial actions determine the requirements for fault isolation. It is not necessary to isolate faults below the level where the fault effect propagation can be stopped. When reconfiguration is needed, and complete isolation can not be achieved within the required time to reconfigure, the set (Σ, τ) will need to be selected assuming a worst-case condition among the set $\{\hat{S}, \hat{\theta}\}$, the available output from the fault diagnosis. The worst case fault is one that has the highest degree of severity.

6. *Design of remedial actions:* Actions are designed to achieve the required fault-tolerance. This can involve simple selection of control level algorithms, enabling and disabling of redundant hardware, or activation/deactivation/replacement of the entire controller. Advanced accommodation can require controller redesign or optimization.

7. *Fault diagnosis design:* The structure information again provides a list of possibilities. The reconfigurability measure for the faulty system indicates how difficult reconstruction will be.

8. *Supervisor logic*: Supervisor inference rules are designed using the information about which faults/effects are detected and how they are treated. The autonomous supervisor determines the most appropriate action from the present condition and commands. The autonomous supervisor must be designed to treat mode changes of the controlled process and any overall/operator commands. Worst-case conditions and overall safety objectives should have priority when full isolation or controller-redesign can not be accomplished within the required time to get within control specifications after a fault.

9. *Test*: Should be complete. The main obstacle is the complexity of the resulting hybrid system consisting of controller and plant. Transient conditions should be carefully tested.

These steps are followed to make the supervisor design cheaper, faster, and better. The fault coverage is then (hopefully) as complete as possible, because the FPA step in principle includes all possible faults. The analysis is modular, because small sub-systems are treated individually. Furthermore, the strategy has the advantage that the system is analyzed on a logical level as far as possible before the laborious job of mathematical modelling and design is initiated. This ensures that superfluous analysis and design are avoided.

7 An Example: Ship Propulsion

To illustrate the methods of analysis, we consider a ship propulsion system , which was defined as a COSY benchmark on fault detection and fault-tolerant control [18], [19].

This example considers a subset of the benchmark to illustrate selected parts of the overall analysis.

7.1 Constraints

Developed thrust and torque are functions of pitch u_2, shaft speed x_1 and ship speed x_2.

Measurements are

$$\begin{array}{ll} f_1: & y_1 = x_1 \\ f_2: & y_2 = x_2 \\ f_3: & u_{1m} = u_1 \\ f_4: & u_{2m} = u_2 \end{array} \tag{13}$$

Diesel engine and dynamic shaft equation

$$\begin{array}{ll} f_5: & K_y = K_{y,nom} \\ f_6: & Q_{eng} = K_y u_1 \\ f_7: & I_t \dot{x}_1 = -Q_{prop} + Q_{eng} \end{array} \tag{14}$$

Propeller and hull

$$\begin{array}{ll} f_8: & Q_{\text{prop}} = Q_{n|n||\vartheta|}\,|u_2|\,|x_1|\,x_1 + Q_{|n|u\vartheta}u_2\,|x_1|\,x_2 \\ f_9: & T_{\text{prop}} = T_{|n|n\vartheta}u_2\,|x_1|\,x_1 + T_{nu}x_1x_2 \end{array} \tag{15}$$

Ship speed and hull resistance,

$$\begin{array}{ll} f_{10}: & m\dot{x}_2 = -R(x_2) + (1-t)T_{prop} \\ f_{11}: & R(x_2) = X_{|\mathrm{u}|u}\,|x_2|\,x_2 \end{array} \tag{16}$$

The differentials $\dot{x}_1$ and $\dot{x}_2$ are the integrals of x_1 and x_2, respectively. Since integration of the derivative can not determine the related state variable, due to unknown initial value, the arrows in the structure diagram are unidirectional.

$$\begin{array}{ll} f_{12}: & \dot{x}_1 = \frac{d}{dt}x_1 \\ f_{13}: & \dot{x}_2 = \frac{d}{dt}x_2 \end{array} \tag{17}$$

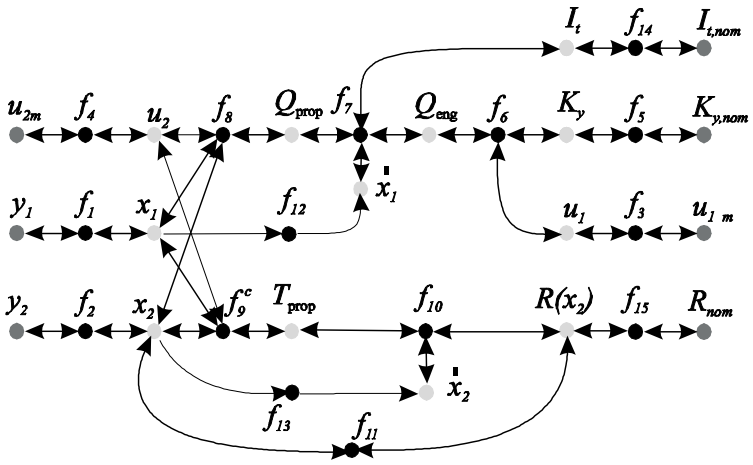

Figure 4: The system-structure graph for the ship propulsion example. Constraints are functional relations between parameters and variables.

This illustrates the difference between observability and calculability, as defined in structure analysis. Finally, parameters are assumed known. All such parameters should have identity constraints associated with them. For brevity, only two are shown in the Figure,

$$\begin{array}{ll} f_{14}: & I_t = I_{t,nom} \\ f_{15}: & R(\cdot) = R(\cdot)_{nom} \end{array} \tag{18}$$

The control objective is to obtain desired ship speed while meeting constraints on the shaft's angular speed. The system's structure graph is shown in Figure **??**

The ship benchmark deals with several faults. One of those is a sensor fault in shaft speed measurement. This means constraint f_1 is violated. This example investigates which redundancy relations exist to reconstruct this measurement.

7.2 Analysis of Structure

We first observe which variables belong to the sets F(15 elements), X(12 elements) and K(7 elements):

$$F = \{f_1, f_2 ... f_{15}\} \tag{19}$$

$$K = \{y_1, y_2, u_{m1}, u_{m2}, K_{y,nom}, I_{t,nom}, R_{nom}\} \tag{20}$$

$$X = \{x_1, \dot{x}_1, x_2, \dot{x}_2, u_1, u_2, Q_{eng}, Q_{prop}, T_{prop}, K_y, I_t, R(\cdot)\} \tag{21}$$

It is noted that $F_K = \emptyset$ as there is no direct redundancy in this system.

A structural analysis of the system gives that the set A of Definition 11 could be ordered as one matrix of dimension $(dim(F), (dim(X) + dim(K)))$. Alternatively, individual entries are listed, and sparse matrix techniques could be used in the analysis,

$$\begin{gathered} A = \{(f_1, y_1), (f_1, x_1), \\ (f_2, y_2), (f_2, x_2), \\ (f_3, u_{1m}), (f_3, u_1), \\ (f_4, u_{2m}), (f_4, u_2), \\ (f_5, K_y), (f_5, K_{y,nom}), \\ (f_6, Q_{eng}), (f_6, K_y), (f_6, u_1), \\ (f_7, Q_{prop}), (f_7, Q_{eng}), (f_7, I_t), (f_7, n_1), \\ (f_8, u_2), (f_8, x_1), (f_8, x_2), (f_8, Q_{prop}), \\ (f_9, x_2), (f_9, u_2), (f_9, x_1), (f_9, T_{prop}), \\ (f_{10}, \dot{x}_2), (f_{10}, R(x_2)), (f_{10}, T_{prop}), \\ (f_{11}, R), (f_{11}, x_2), \\ (f_{12}, \dot{x}_1), (f_{12}, x_1), \\ (f_{13}, \dot{x}_2), (f_{13}, x_2), \\ (f_{14}, I_t), (f_{14}, I_{t,nom}), \\ (f_{15}, R), (f_{15}, R_{nom}),\} \end{gathered} \tag{22}$$

From Definition 12, the set of sub-systems that exist, comprise sets of selected constraints with their associated variables, and combinations of such sets, are

$$\begin{aligned} \varphi = & \{(f_1, y_1, x_1), ..., (f_{11}, R(x_2), x_2, R_{nom}), ..\} \\ & \cup \{(f_1, f_2, y_1, x_1, y_2, x_2), ...\} \\ & \cup \{(f_i, f_j, y_k, ...), ...\} ... \end{aligned} \tag{23}$$

Several complete matchings on X exist, and some of these are listed in table 1. In the table, the elements refer to the constraints (as elements in A) that are associated with each variable in X. In match 1, x_1 is associated with f_1 whereas it is associated with f_8 in matching number 2.

In the non-faulty case, x_1 is assessed through measurement, formally through constraint 1 and the arc $(f_1, x_1, (f_1, y_1))$. If the fault in f_1 occurs, analytic redundancy relations should be found that reconstruct x_1 from other relations. This is possible from matchings 2, 3 or 4, which do not include f_1. In constructing the analytic redundancy relations, one has to consider the causality, it is not possible to calculate x_1 from $\dot{x}_1$ through f_{12} as listed in matching 3, since the initial value at the start of calculation is unknown.

Observer techniques could, nevertheless, be employed to provide a useful - and asymptotically correct - estimate, if observation was started well in advance of the fault incident.

The conclusion is that in the faulty case, we could use two remaining ARRs, one uses constraint f_8, the other f_9.

X-element	Match 1	Match 2	Match 3	Match 4
x_1	1	8	12	9
$\dot{x}_1$	7	12	7	12
x_2	2	13	2	2
$\dot{x}_2$	10	10	13	13
u_1	3	3	3	3
u_2	4	4	4	4
Q_{eng}	6	6	6	6
Q_{prop}	8	7	8	7
T_{prop}	9	9	10	10
K_y	5	5	5	5
I_t	14	14	14	14
$R(\cdot)$	15	15	11	11

Table 1: Four examples on complete matching on X for the example

7.2.1 Recovery from the sensor fault

The linearized state equations for this example are derived around a point of operation $(\bar{x}_1, \bar{x}_2)$. A non-dimensionalisation is obtained by $x\prime_1 = x_1/x_{1,\max}$, etc. The parameters are $\theta_1 = -Q_{nn\vartheta}/I_t, \theta_2 = -Q_{nu\vartheta}/I_t, \quad \theta_3 = K_y/I_t, \quad \theta_4 = X_{\mathrm{u}\,u}/m, \quad \theta_5 = (1-t)T_{nn}/m, \quad \text{and } \theta_6 = -(1-t)T_{nu\vartheta}/m.$ The non-dimensional parameters are $\theta'_1 = \theta_1 \frac{x^2_{1,\max}}{\dot{x}_{1,\max}}$ etc.

The non-dimensional linearized state equation is then

$$\begin{bmatrix} \dot{x}'_1 \\ \dot{x}'_2 \end{bmatrix} = \begin{bmatrix} 2\theta'_1\bar{x}'_1\bar{u}'_2 + \theta'_2\bar{x}'_2\bar{u}'_2 & \theta'_2\bar{x}'_1\bar{u}'_2 \\ 2\theta'_5\bar{x}'_1 + \theta'_6\bar{x}'_2\bar{u}_2 & \theta'_6\bar{x}'_1\bar{u}_2 + 2\theta'_4\bar{x}'_2 \end{bmatrix} \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} + \begin{bmatrix} \theta'_3 & \theta'_1\bar{x}'^2_1 + \theta'_2\bar{x}'_1\bar{x}'_2 \\ 0 & \theta'_6\bar{x}'_1\bar{x}'_2 \end{bmatrix} \begin{bmatrix} u'_1 \\ u'_2 \end{bmatrix} \tag{24}$$

The output for the non-faulty system is

$$\begin{bmatrix} y'_{1m} \\ y'_{2m} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x'_{1m} \\ x'_{2m} \end{bmatrix} \tag{25}$$

The output for the faulty system is

$$[y'_{2m}] = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x'_{1m} \\ x'_{2m} \end{bmatrix} \tag{26}$$

The quality measure based on observability index is $\rho_o = 0.07$. This shows that recovery from the shaft speed fault will be possible, but a large part of observation "energy" is lost when the sensor fault occurs.

8 Summary

This paper has introduced fault-tolerant control as a new discipline within automatic control. The objective was to increase plant availability and reduce

the risk of safety hazards when faults occur. Concise definitions were given to cover the hierarchy from fault-tolerant control to supervision, with the remedial actions to faults being defined as accommodation and reconfiguration depending on the degree of redundancy in the controlled process. Principal analysis of essential system properties were treated, the topics were selected to give the essence of an overall fault tolerant design. These included fault propagation analysis, structural analysis and selection of the best remedial actions based on measures of recovery for a system when a particular fault occurs. An example from a ship propulsion benchmark was used to show salient features of different parts of the design.

References

[1] M. Blanke, *Consistent design of dependable control systems*, Control Engineering Practice **4** (1996), no. 9, 1305–1312.

[2] M. Blanke, S. A. Bøgh, R. B. Jørgensen, and R. J. Patton, *Fault detection for a diesel engine actuator - a benchmark for fdi*, Control Engineering Practice **3** (1995), 1731–1740.

[3] M. Blanke, O. Borch, G. Allasia, and F. Bagnoli, *Development of and automated technique for failure modes and effects analysis*, Safety and Reliability, Proc. Of ESREL'99 - the Tenth European Conf. On Safety and Reliability (Rotterdam) (G. I. Schueller and P. Kafka, eds.), A. A. Balkema, September 1999, pp. 839–844.

[4] M. Blanke, R. Izadi-Zamanabadi, S. A. Bøgh, and C. P. Lunau, *Fault-tolerant control systems - a holistic view*, Control Engineering Practice **5** (1997), no. 5, 693–702.

[5] M. Blanke, R. Izadi-Zamanabadi, and T. F. Loostma, *Fault monitoring and re-configurable control for a ship propulsion plant*, Journal of Adaptive Control and Signal Processing (1998), 671–688.

[6] S. A. Bøgh, *Fault tolerant control systems - a development method and real-life case study*, Ph.D. thesis, Dept. of Control Eng., Aalborg University, Denmark, December 1997.

[7] S. A. Bøgh, R. Izadi-Zamanabadi, and M. Blanke, *Onboard supervisor for the ørsted satellite attitude control system*, Artificial Intelligence and Knowledge Based Systems for Space, 5th Workshop (Noordwijk , Holand), The European Space Agency, Automation and Ground Facilities Division, October 1995, pp. 137–152.

[8] V. Cocquempot, J. Ph. Cassar, and M. Staroswiecki, *Generation of robust analytical redundancy relations*, Proceedings of ECC'91, Grenoble, France, July 1991, pp. 309-314.

[9] V. Cocquempot, R. Izadi-Zamanabadi, M. Staroswiecki, and M. Blanke, *Residual generation for the ship benchmark using structural approach*, IEE Control'98 (Swansea, UK), September 1998.

[10] P. Declerck and M. Staroswiecki, *Characterization of the canonical components of a structural graph for fault detection in large scale industrial plants*, Proceedings of ECC'91 (Grenoble, France), July 1991, pp. 298–303.

[11] A. L. Dulmage and N. S. Mendelsohn, *Coverings of bipartite graphs*, 1958, pp. 517–534.

[12] C. W. Frei, F. J. Kraus, and M. Blanke, *Recoverability viewed as a system property*, Proc. European Control Conference 1999, ECC'99, September 1999.

[13] A. L. Gehin and M. Staroswiecki, *A formal approach to reconfigurability analysis - application to the three tank benchmark*, Proc. European Control Conference 1999, ECC'99, September 1999.

[14] K. Glover and L.M. Silverman, *Characterization of structural controllability*, **31** (1976), 534–537.

[15] E. J. Henley and R. A. Williams, *Graph theory in modern engineering*, Academic Press, New York, 1973.

[16] S. A. Herrin, *Maintainability applications using the matrix fmea technique*, Transactions on Reliability **R-30** (1981), no. 2, 212–217.

[17] R. Isermann and P. Ballé, *Trends in the application of model-based fault detection and diagnosis of technical processes*, Control Engineering Practice **5** (1997), no. 5, 709–719.

[18] R. Izadi-Zamanabadi and M. Blanke, *A ship propulsion system as a benchmark for fault-tolerant control*, Control Engineering Practice **7** (1999), no. 2, 227–239.

[19] Roozbeh Izadi-Zamanabadi, *Fault-tolerant supervisory control - system analysis and logic design*, Ph.D. thesis, Dept. of Control Eng., Aalborg University, Denmark, September 1999.

[20] D. Bonvin J. P. Keller, *Selection of input and output variables as a model reduction problem*, Automatica **28** (1992), no. 1, 171–177.

[21] J. M. Legg, *Computerized approach for matrix-form fmea*, IEEE Transactions on Reliability **R-27** (1978), no. 1, 254–257.

[22] Chen-Tai Lin, *Structural controllability*, IEEE Trans. Automat. Contr. **AC-19** (1974), no. 3, 201–208.

[23] Morten Lind, *Modeling goals and functions of complex industrial plants*, Applied Artificial Intelligence **8** (1994), 259–283.

[24] Charlotte P. Lunau, *A reflective architecture for process control applications.*, ECOP'97 Object Oriented Programming (M. Aksit and S. Matsuoka, eds.), Springer Verlag, 1997, Lecture Notes in Computer Science, Vol. 1241, pp. 170–189.

[25] J. Lunze and J. Schröder, *Process diagnosis based on a discrete–event description*, Automatisierungstechnik **47** (1999), no. 8.

[26] Jan Lunze, *Qualitative modelling of linear dynamical systems with quantized state measurements*, Automatica **30** (1994), no. 3, 417–431.

[27] Jan Lunze and F. Schiller, *Logic-based process diagnosis utilising the causal structure of dynamical systems*, Preprints of IFAC/IFIP/IMACS Int. Sympo. on Artificial Intelligence in Real-time Control: AIRTC'92 (Delft), Jun. 16-18 1992, pp. 649–654.

[28] A. Misra, *Sensor-based diagnosis of dynamical systems*, Ph.D. thesis, Vanderbilt University, 1994.

[29] M. Morari and G. Stephanopoulos, *Studies in the synthesis of control structures for chemical processes. Part II: Structural aspects and the synthesis of alternative feasible control schemes*, AIChE Journal **40** (1980), no. 2, 232–246.

[30] P.C. Müller and H.I. Weber, *Analysis and optimization of certain qualities of controllability and observability for linear dynamical systems*, **8** (1972), 237–246.

[31] Ron J. Patton, *Fault tolerant control: The 1997 situation*, IFAC Safeprocess'97 (Hull, United Kingdom), August 1997, pp. 1033–1055.

[32] H. H. Rosenbrock, *State-space and multivariable theory*, Thomas Nelson and Sons LTD., 1970.

[33] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. C. Teneketzis, *Failure diagnosis using discrete-event models*, IEEE Trans. on Control Systems Techn. **4** (1996), no. 2, 105–123.

[34] M. Staroswiecki, S. Attouche, and M. L. Assas, *A graphic approach for reconfigurability analysis*, Proc. DX'99, June 1999.

[35] M. Staroswiecki and P. Declerck, *Analytical redundancy in non-linear interconnected systems by means of structural analysis*, vol. II, IFAC-AIPAC'89, July 1989, pp. 23–27.

[36] M. Staroswiecki and A. L. Gehin, *Analysis of system reconfigurability using generic component models*, CONTROL'98, September 1998, pp. 1157–1162.

[37] ———, *Control, fault tolerant control and supervision problems*, IFAC Int. Symposium on Safety in Technical Processes Safeprocess' 2000 /it-submitted (Budapest, Hungary), 2000.

[38] Marcel Staroswiecki and Mireille Bayart, *Models and languages for the interoperability of smart instruments*, Automatica **32** (1996), no. 6, 859–873.

[39] Jakob Stoustrup and M. J. Grimble, *Integrating control and fault diagnosis: A separation result*, IFAC Sym. on Fault Detection, Supervision and Safety for Technical Processes (Hull, United Kingdom), August 1997, pp. 323–328.

[40] P. M. Frank T. Marcu, M. H. Matcovschi, *Neural approaches to observer-based fault diagnosis and reconfiguration of a three-tank system*, COSY Workshop (Mulhouse, France), April 3-4, 1998.

[41] R. J. Veillette, J. V. Medani, and W. R. Perkins, *Design of reliable control systems*, Trans. on Automatic Control **37** (1992), no. 3, 290–304.

[42] J.L. Willems, *Structural controllability and observability*, Syst. Control Lett. (1986), 5–12.

[43] W. M. Wonham, *A control theory for discrete-event system*, Advanced Computing Concepts and Techniques in Control Engineering (M.J. Denham and A.J. Laub, eds.), Springer-Verlag, 1988, pp. 129–169.